

# R-Skript PUNO-Forschungsprojekt

## Teil 3 – Tabellen nach Word exportieren

*Dominik Vogel*

*Stand: 04.06.2019*



## Inhaltsverzeichnis

1	Vorbemerkungen	1
2	Pakete	1
3	Grundlagen von <code>stargazer</code>	1
4	Deskriptive Kennwerte	3
5	Korrelationstabelle	3
6	Regressionstabelle	5

## 1 Vorbemerkungen

R wird idealtypisch nicht mit Word sondern mit  $\text{\LaTeX}$  verwendet. Viele Funktionen zum Export sind daher darauf ausgelegt, Ergebnisse in  $\text{\LaTeX}$  zu exportieren. Wir verwenden hier daher eine Variante, die eigentlich für den Export für Internetseiten gedacht ist. Die vorgestellten Funktionen produzieren zunächst eine HTML-Datei. Diese Datei öffnen wir anschließend mit Word und kopieren die Tabelle in ein normales Worddokument.

## 2 Pakete

Wir benötigen zunächst wieder das `tidyverse` Paket:

```
library(tidyverse)
```

Für den Export verwenden wir außerdem das Paket `stargazer`

```
install.packages("stargazer", dep = TRUE)
library(stargazer)
```

## 3 Grundlagen von `stargazer`

`stargazer` ist darauf ausgelegt, einen Data Frame oder verschiedene Modelle, wie zum Beispiel ein Regressionsmodell, zu verarbeiten. `stargazer` erstellt dann eine formatierte Tabelle als reine Textausgabe (`type = "text"`), als  $\text{\LaTeX}$ -Code (`type = "latex"`) oder Website (`type = "html"`).

Unter <https://cran.r-project.org/web/packages/stargazer/vignettes/stargazer.pdf> findet sich eine ausführliche Anleitung. In der Hilfe (`?stargazer`) gibt es außerdem eine Übersicht über die vielfältigen Anpassungsmöglichkeiten.

Starten wir als kurzes Beispiel mit einer Tabelle deskriptiver Kennwerte. Hier müssen wir einen kleinen Trick anwenden, da `stargazer` nicht mit dem speziellen Format von Data Frame umgehen kann, das wir mit `read_csv()` bekommen (Tibble). Wir setzen den Datensatz `albums` daher in die Funktion `as.data.frame`.

```
albums <- read_csv("data/album_sales.csv") # Albumdaten aus Teil 2 importieren
```

```
## Parsed with column specification:
## cols(
##   adverts = col_double(),
##   sales = col_double(),
##   airplay = col_double(),
##   attract = col_double(),
##   hiphop = col_double()
## )
```

```
stargazer(as.data.frame(albums),
          type = "html")
```

```
##
## <table style="text-align:center"><tr><td colspan="8" style="border-bottom: 1px solid black"></td></tr><tr><td colspan="8" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:le
## <tr><td style="text-align:left">sales</td><td>200</td><td>194,700.000</td><td>85,171.480</td><td>
## <tr><td style="text-align:left">airplay</td><td>200</td><td>27.500</td><td>12.270</td><td>0</td><td>
## <tr><td style="text-align:left">attract</td><td>200</td><td>6.770</td><td>1.395</td><td>1</td><td>
## <tr><td style="text-align:left">hiphop</td><td>200</td><td>0.595</td><td>0.492</td><td>0</td><td>
## <tr><td colspan="8" style="border-bottom: 1px solid black"></td></tr></table>
```

Wir sehen, dass `stargazer` standardmäßig den reinen HTML-Code ausgibt. Damit können wir allerdings nicht viel anfangen. Wir leiten den Output daher mit der Option `out="deskriptiv.html"` in eine Datei um.

```
stargazer(as.data.frame(albums),
          type = "html",
          out = "deskriptiv.html")
```

Das Resultat sieht ungefähr so aus:

	Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
adverts	200	520,127.400	457,559.000	1	102,623.5	828,361.5	1,985,119	
sales	200	194,700.000	85,171.480	10,000	137,500	250,000	590,000	
airplay	200	27.500	12.270	0	19.8	36	63	
attract	200	6.770	1.395	1	6	8	10	
hiphop	200	0.595	0.492	0	0	1	1	

## 4 Deskriptive Kennwerte

Nachdem wir nun das Grundprinzip verstanden haben, können wir die Standardausgabe noch etwas anpassen. Wir beschränken hierfür die Zahl der Nachkommastellen auf 2, benennen die Variablen um und fügen eine Beschriftung hinzu:

```
stargazer(as.data.frame(albums),
  type = "html",
  out = "deskriptiv2.html",
  digits = 2, # Anzahl der Nachkommastellen auf 2 beschränken
  digits.extra = 2,
  covariate.labels = c("Werbeausgaben in Pfund",
    "Verkaufszahlen",
    "Anzahl Wiedergaben Radio",
    "Attraktivität Band",
    "Hip-Hop (1 = ja)",
  title = "Deskriptive Statistik")
```

Das Resultat:

Deskriptive Statistik							
Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
Werbeausgaben in Pfund	200	520,127.40	457,559.00	1	102,623.5	828,361.5	1,985,119
Verkaufszahlen	200	194,700.00	85,171.48	10,000	137,500	250,000	590,000
Anzahl Wiedergaben Radio	200	27.50	12.27	0	19.8	36	63
Attraktivität Band	200	6.77	1.40	1	6	8	10
Hip-Hop (1 = ja)	200	0.60	0.49	0	0	1	1

## 5 Korrelationstabelle

Für eine Korrelationstabelle übergeben wir einfach das Resultat aus `cor()` an `stargazer`

```
cor_albums <- cor(albums,
  use = "pairwise.complete.obs") # wegen NAs
```

```
stargazer(cor_albums,
  type = "html",
  out = "cor.html")
```

	adverts	sales	airplay	attract	hiphop
adverts	1	0.368	0.111	0.021	0.107
sales	0.368	1	0.604	0.337	0.279
airplay	0.111	0.604	1	0.182	0.101
attract	0.021	0.337	0.182	1	0.208
hiphop	0.107	0.279	0.101	0.208	1

Um die Korrelationstabelle noch etwas sparsamer zu machen, streichen wir die doppelten Informationen. Hierzu müssen wir allerdings ein wenig Hand anlegen.

```
# Hide upper triangle
upper <- cor_albums # neues Objekt erstellen
upper[upper.tri(cor_albums)] <- NA # Werte für obere Triangel löschen

# letzte Spalte entfernen
limit <- ncol(upper) - 1 # Letzte Spalte ermitteln
upper <- upper[,1:limit] # letzte Spalte ausblenden

# In Data Frame umwandeln
upper <- data.frame(upper)
```

Bevor wir die Tabelle erstellen können, müssen wir zunächst noch die Zeilen benennen. Hierfür nutzen wir die Funktion `setattr()` aus dem Paket `data.table`.

```
install.packages("data.table", dep = TRUE)
library(data.table)
```

Wir vergeben neben den Variablennamen noch fortlaufende Nummern, um im Spaltenkopf der finalen Tabelle etwas Platz zu sparen:

```
upper <- (setattr(upper, "row.names", c("(1) Werbeausgaben in Pfund",
                                         "(2) Verkaufszahlen",
                                         "(3) Anzahl Wiedergaben Radio",
                                         "(4) Attraktivität Band",
                                         "(5) Hip-Hop (1= ja)")))
```

Nun können wir endlich die Tabelle erstellen:

```
# Tabelle mit stargazer erstellen
stargazer(upper,
  type = "html",
  out = "cor2.html",
  summary = FALSE, # Deskriptive Kennwerte verhindern
  digits = 2, # Anzahl der Nachkommastellen auf 2 beschränken
  digits.extra = 2,
  covariate.labels = c("", "(1)", "(2)", "(3)", "(4)") # Spalten benennen
)
```

	(1)	(2)	(3)	(4)
(1) Werbeausgaben in Pfund	1			
(2) Verkaufszahlen	0.37	1		
(3) Anzahl Wiedergaben Radio	0.11	0.60	1	
(4) Attraktivität Band	0.02	0.34	0.18	1
(5) Hip-Hop (1= ja)	0.11	0.28	0.10	0.21

## 6 Regressionstabelle

Für die Regressionstabelle müssen wir glücklicherweise weniger Aufwand betreiben. Wir schätzen zunächst die Modelle die wir in die Tabelle integrieren wollen und übergeben diese anschließend an `stargazer`.

```
model_albums1 <- lm(sales ~ adverts, data = albums)
model_albums2 <- lm(sales ~ adverts + airplay + attract, data = albums)
stargazer(model_albums1, model_albums2,
           type = "html",
           out = "reg.html")
```

	<i>Dependent variable:</i>	
	sales	
	(1)	(2)
adverts	0.068*** (0.012)	0.057*** (0.009)
airplay		3,663.286*** (355.094)
attract		14,332.880*** (3,103.912)
Constant	159,083.700*** (8,512.134)	-32,534.280 (22,224.420)
Observations	200	200
R <sup>2</sup>	0.135	0.510
Adjusted R <sup>2</sup>	0.131	0.503
Residual Std. Error	79,398.850 (df = 198)	60,074.010 (df = 196)
F Statistic	30.988*** (df = 1; 198)	68.003*** (df = 3; 196)
<i>Note:</i> *p<0.1; **p<0.05; ***p<0.01		

Das Ergebnis kann sich schon sehen lassen. Wir nehmen nur noch ein paar Anpassungen vor:

```
stargazer(model_albums1, model_albums2,
  type = "html",
  out = "reg2.html",
  align = TRUE, # Werte untereinander ausrichten
  title = "Regression results for explaining album sales", # Titel
  dep.var.caption = "", # "Dependent variable" ausblenden
  dep.var.labels.include = FALSE, # Keine Angabe der abhängigen Variable
  covariate.labels = c("Werbeausgaben in Pfund", # uaV benennen
    "Verkaufszahlen",
    "Anzahl Wiedergaben Radio",
    "Attraktivität Band",
    "Hip-Hop (1 = ja)" ),
  omit.stat = c("ser"), # Standardfehler der Residuen ausblenden
  star.cutoffs = c(0.05, 0.01, 0.001), # Sternchen für 0.05, 0.01, 0.001
  notes.append = TRUE, # Standardnotiz ergänzen
  notes = c("standard errors in parentheses") # Note anpassen
)
```

Nun haben wir eine publikationsfähige Regressionstabelle

Regression results for explaining album sales		
	(1)	(2)
Werbeausgaben in Pfund	0.068*** (0.012)	0.057*** (0.009)
Verkaufszahlen		3,663.286*** (355.094)
Anzahl Wiedergaben Radio		14,332.880*** (3,103.912)
Attraktivität Band	159,083.700*** (8,512.134)	-32,534.280 (22,224.420)
Observations	200	200
R <sup>2</sup>	0.135	0.510
Adjusted R <sup>2</sup>	0.131	0.503
F Statistic	30.988*** (df = 1; 198)	68.003*** (df = 3; 196)
Note:	* p<0.05; ** p<0.01; *** p<0.001 standard errors in parentheses	